CLAIMS

1. A computer program product, tangibly embodied in an information carrier, comprising instructions operable to cause data processing apparatus to:

implement a reusable software component encapsulating functionality, multiple instances of the component being usable at the same time;

the component having at least one visual representation;

the component having a programming interface for programmatic interaction with the component;

the component having a data-binding interface for data communication with the component; and

the component having a visual interface for access to the at least one visual representation of the component.

2. The computer program product of claim 1, wherein the programming interface, the data-binding interface, and the visual interface are separate interfaces.

3. The computer program product of claim 1, further comprising instructions to implement one or more controllers for the component, each controller having an associated context for storing data and state for the controller.

4. The computer program product of claim 3, further comprising instructions to implement one or more views for the component, each view providing a visual representation of the component.

5. The computer program product of claim 1, further comprising instructions to:

embed a sub-component into the component.

6. The computer program product of claim 5 wherein the instructions to embed a sub-component comprise instructions to:

use a programming interface, a data-binding interface, and a visual interface of the sub-component.

7. The computer program product of claim 1 wherein the programming interface includes an interface controller having an interface controller context and a configuration controller having a configuration controller context, the visual interface includes an interface view, and the data-binding interface provides context mapping for the interface controller context and the configuration context.

8. A computer program product, tangibly embodied in an information carrier, for implementing an application runtime framework, the computer program product comprising instructions operable to cause data processing apparatus to:

receive a component interface to be used in an application without a specification of a corresponding component implementation; and

instantiate a particular component implementation at application runtime, the particular component implementation being selected from one or more component implementations corresponding to the component interface.

9. The computer program product of claim 8, wherein the component interface has a programming interface, a data-binding interface, and a visual interface.

10. A computer program product, tangibly embodied in an information carrier, for implementing an application runtime framework, the computer program comprising instructions operable to cause data processing apparatus to:

receive an event subscription for a subscribing component, the event subscription specifying subscriptions to one or more events generated by sub-components embedded by the subscribing component;

cache any events generated by the sub-components that are specified by the event subscription if the subscribing component has not been instantiated; and

forward any cached events to an instance of the subscribing component after the subscribing component is instantiated.

11. A computer program product, tangibly embodied in an information carrier, for implementing an application runtime framework, the computer program product comprising instructions operable to cause data processing apparatus to:

receive one or more context mappings for a component, the context mappings being specified by a component embedder using the component to exchange context data with the component;

if the component has not been instantiated, cache the specified context mappings; and

create the specified context mappings for the component after the component has been instantiated.

12. A computer implemented method, comprising:

implementing a reusable software component encapsulating functionality, multiple instances of the component being usable at the same time;

the component having at least one visual representation;

the component having a programming interface for programmatic interaction with the component;

the component having a data-binding interface for data communication with the component; and

the component having a visual interface for access to the at least one visual representation of the component.

13. The method of claim 12, wherein implementing the component comprises:

implementing the component having the programming interface, the data-binding interface, and the visual interface as separate interfaces.

14. The method of claim 12, further comprising:

implementing one or more controllers for the component, each controller having an associated context for storing data and state for the controller.

15. The method of claim 14, further comprising:

implementing one or more views for the component, each view providing a visual representation of the component.

16. The method of claim 12, further comprising:

embedding a sub-component into the component.

17. The method of claim 16, wherein embedding the sub-component comprises:

using a programming interface, a data-binding interface, and a visual interface of the sub-component.

18. The method of claim 12, wherein implementing the component comprises:

implementing the component having the programming interface including an interface controller having an interface controller context and a configuration controller having a configuration controller context, the visual interface including an interface view, and the data-binding interface providing context mapping for the interface controller context and the configuration context.

19. An apparatus, comprising:

means for implementing a reusable software component encapsulating functionality, multiple instances of the component being usable at the same time;

the component having at least one visual representation;

the component having a programming interface for programmatic interaction with the component;

the component having a data-binding interface for data communication with the component; and

the component having a visual interface for access to the at least one visual representation of the component.